I Erläuterungen

Voraussetzungen gemäß KCBG und Abiturerlassen BG jeweils in der für den Abiturjahrgang geltenden Fassung

Standardbezug
Die nachfolgend ausgewiesenen Kompetenzbereiche sind für die Bearbeitung der jeweiligen Aufgabe besonders bedeutsam. Darüber hinaus können weitere, hier nicht ausgewiesene Kompetenzbereiche für die Bearbeitung der Aufgabe nachrangig bedeutsam sein, zumal die Kompetenzbereiche in engem Bezug zueinander stehen. Die Operationalisierung des Bezugs zu den Kompetenzbereichen des Standardbezugs erfolgt in Abschnitt II.

Aufgabe	Kompetenzbereiche				
	K1	K2	K3	K4	K5
1.1			X		
1.2.1	X				
1.2.2					X
1.2.3			X		
1.3				X	
1.4			X		
2.1			X		
2.2		X			
2.3				X	
3.1	X				
3.2	X				
3.3		X			

Inhaltlicher Bezug
Die nachfolgend ausgewiesenen Themenfelder sind die wesentliche inhaltliche Grundlage für die vorliegenden Aufgaben. Darüber hinaus können weitere, hier nicht explizit ausgewiesene Themenfelder für die Bearbeitung nachrangig bedeutsam sein.
Q1: Objektorientierte Softwareentwicklung
Q2: Digitale Steuerungstechnik
Q3: Prozessautomatisierung
verbindliche Themenfelder: Objektmodellierung (Q1.1), Implementierung von Klassen und ihren Beziehungen (Q1.2), Synthese statischer und sequentieller Logikschaltungen (Q2.1), Mikrocontroller (Q2.2)

II Lösungshinweise

In den nachfolgenden Lösungshinweisen sind alle wesentlichen Gesichtspunkte, die bei der Bearbeitung der einzelnen Aufgaben zu berücksichtigen sind, konkret genannt und diejenigen Lösungswege aufgezeigt, welche die Prüflinge erfahrungsgemäß einschlagen werden. Selbstverständlich sind jedoch Lösungswege, die von den vorgegebenen abweichen, aber als gleichwertig betrachtet werden können, ebenso zu akzeptieren.

Nicht für den Prüfling bestimmt

Aufg.	erwartete Leistungen	BE I	BE II	BE III
1.1	entwickeln, bestimmen entwickeln bestimmen	 4	4 	
1.2.1	erläutern, erklären Die Klasse `Experimentierkastenliste` ist die Container-Klasse der Klasse `Experimentierkasten`, d.h. ihre einzige Aufgabe ist es, die Objekte der Klasse `Experimentierkasten` zu verwalten. Im statischen Attribut `instance` kann eine Referenz auf ein Experimentierkastenlistenobjekt gespeichert werden. Somit kann überprüft werden, ob es bereits ein Experimentierkastenlistenobjekt gibt, da es nur ein Objekt der Container-Klasse geben darf. erläutern erklären		2 2	
1.2.2	beschreiben, erklären Zwischen den Klassen `Bestellung` und `Bestellposition` besteht eine Kompositions-Beziehung, d.h. jede Bestellposition ist Teil genau einer Bestellung. Wird die Bestellung gelöscht, so werden auch die Bestellpositionen gelöscht. Jede Bestellung besteht aus mindestens einer Bestellposition. Die Klasse `Bestellposition` wurde im UML-Klassendiagramm modelliert, um die Anzahl der Exemplare eines bestellten Bauteils in einem Attribut erfassen zu können. Diese kann weder direkt in der Klasse `Bestellung` noch in der Klasse `Bauteil` gespeichert werden. beschreiben erklären	3 		 2

Aufg.	**erwartete Leistungen**	**BE**		
		I	**II**	**III**
1.2.3	ergänzen, modellieren **Bestellposition** - anzahl : int + <<create>> Bestellposition(t : Bauteil, anzahl : int) 1..* * ◀ bezieht sich auf 1 **Experimentierkasten** - artikelnr : int - name : String - kastenpreis:double : int + <<create>> Experimentierkasten(name : String, mindestalter : int, kastenpreis : double) + getMindestalter() : int + getKastenpreis() : double * 1..* besteht aus ▶ * **Bauteil** - teilenummer : int - bezeichnung : String - preis : double + <<create>> Bauteil(bezeichnung : String, preis : double)			
	ergänzen	2		
	modellieren		2	2

Nicht für den Prüfling bestimmt

Nicht für den Prüfling bestimmt

Aufg.	erwartete Leistungen	BE I	BE II	BE III
1.3	implementieren		5	13

```
import java.util.ArrayList;

public class Experimentierkastenliste{

private static Experimentierkastenliste instance =null;
private ArrayList<Experimentierkasten> liste= new Ar-
rayList<Experimentierkasten>();

private Experimentierkastenliste() {}

public static Experimentierkastenliste getInstance() {
             if (instance==null) {
                   instance=new Experimentierkastenliste();
             }
             return instance;
}

public ArrayList <Experimentierkasten> experimen-
tierkastenAnzeigen (int alter, double preis){
   ArrayList<Experimentierkasten> neu = new ArrayList
   <Experimentierkasten>();
     for(int i=0; i<liste.size();i++) {
        if(liste.get(i).getMindestalter()<=alter &&
        (liste.get(i).getKastenpreis()*1.1)<=preis) {
            neu.add(liste.get(i));
        }
     }
    return neu;
}

public ArrayList<Bauteil> bauteileAnzeigen (Experimentier-
kasten e){
 return e.getTeile();
}

}



import java.util.ArrayList;

public class Experimentierkasten {
      private int artikelnr;
      private String name;
      private int mindestalter;
      private double kastenpreis;
      private ArrayList<Bauteil> teile;
      private static int zähler=1;
```

Aufg.	erwartete Leistungen	BE I	BE II	BE III

	<pre>public Experimentierkasten(String name, int mindestalter,
	double kastenpreis) {
	            this.name=name;
	            this.mindestalter=mindestalter;
	            this.kastenpreis=kastenpreis;
	            teile=new ArrayList<Bauteil>();
	            this.artikelnr=zähler++;
	}
	
	public int getMindestalter() {
	      return mindestalter;
	}
	
	public double getKastenpreis() {
	      return kastenpreis;
	}
	
	public ArrayList<Bauteil> getTeile() {
	      return teile;
	}
	}</pre>			

Nicht für den Prüfling bestimmt

Aufg.	erwartete Leistungen	BE I	BE II	BE III
1.4	modellieren k : Kunde b : Bestellung : Bestellposition x : UI bauteileBestellen(teile, anzahl) : boolean <<create>> Bestellung(k, teile, anzahl) loop [i=0; i<teile.length; i++] <<create>> bp = Bestellposition(teile[i], anzahl[i]) bestellpositionen.add(bp):boolean() bool = bestellungen.add(b):boolean() [bool]		4	6
	Summe 51	**9**	**19**	**23**

Aufg.	erwartete Leistungen	BE I	BE II	BE III
2.1	darstellen Anzahl der Wiederholungen (x) über Port B einlesen solang x>0 Bauteil greifen solange bis Sensor 1 == High zähler = 5 Servo High setzen warten 1,75 ms Servo Low setzen warten 18,25 ms zähler-- solange zähler >0 Befüllen solange bis Sensor 2 == High zähler = 5 Servo High setzen warten 1,5 ms Servo Low setzen warten 18,5 ms zähler-- solange zähler >0 x--	5	7	
2.2	zuordnen − Port B: Pins 0-7 Eingänge für 8-stelliges Signal (Anzahl der Wiederholungen), − Port D: Pin 0 Eingang für Sensor 1, − Port D: Pin 1 Eingang für Sensor 2, − Port D: Pin 2 Ausgang Signal für geregelten Motor.	2		

Aufg.	**erwartete Leistungen**	**BE**		
		I	**II**	**III**

2.3	implementieren
	`in r16, 0x03`
	`wiederholen:`
	`cpi r16,0`
	`breq end`
	`greifen:`
	`call bauteilgreifen`
	`sbis 0x09,0`
	`jmp greifen`
	`ldi r17,5`
	`drehenrechts:`
	`sbi 0x0B,2`
	`call warten1_75ms`
	`cbi 0x0B,2`
	`call warten18_25ms`
	`dec r17`
	`brne drehenrechts`
	`fuellen:`
	`call befuellen`
	`sbis 0x09,1`
	`jmp fuellen`
	`ldi r17,5`
	`drehenlinks:`
	`sbi 0x0B,2`
	`call warten1_5ms`
	`cbi 0x0B,2`
	`call warten18_5ms`
	`dec r17`
	`brne drehenlinks`
	`dec r16`
	`jmp wiederholen`
	`end:`
			6	7

	Summe 27	**7**	**13**	**7**

Nicht für den Prüfling bestimmt

Aufg.	erwartete Leistungen	BE I	BE II	BE III
3.1	zeichnen	5	7	

Nicht für den Prüfling bestimmt

Aufg.	erwartete Leistungen	BE I	BE II	BE III
3.2	zeichnen Taktsignal (1, 0, t) Eingangssignal (1, 0, t) Q_1 (1, 0, t) Q_2 (1, 0, t)	3	3	
3.3	beschreiben Beim seriellen Laden werden die Eingangssignale nacheinander über einen Pin taktgesteuert eingelesen. Beim parallelen Laden werden alle FlipFlops gleichzeitig geladen. Bei der seriellen Ausgabe werden die Signale nacheinander am letzten FlipFlop taktgesteuert ausgegeben. Bei der parallelen Ausgabe werden alle Stellen des Binärworts zeitgleich aus allen FlipFlops ausgegeben.	4		
	Summe 22	**12**	**10**	

III Bewertung und Beurteilung

Die Bewertung und Beurteilung erfolgt unter Beachtung der nachfolgenden Vorgaben nach § 33 der Oberstufen- und Abiturverordnung (OAVO) in der jeweils geltenden Fassung. Bei der Bewertung und Beurteilung der sprachlichen Richtigkeit in der deutschen Sprache sind die Bestimmungen des § 9 Abs. 12 Satz 3 OAVO in Verbindung mit Anlage 9b anzuwenden.

Bei der Bewertung und Beurteilung der Übersetzungsleistung in den Fächern Latein und Altgriechisch sind die Bestimmungen des § 9 Abs. 14 OAVO in Verbindung mit Anlage 9c anzuwenden.

Der Fehlerindex ist nach Anlage 9b zu § 9 Abs. 12 OAVO zu berechnen. Für die Ermittlung der Punkte nach Anlage 9a zu § 9 Abs. 12 OAVO sowie Anlage 9c zu § 9 Abs. 14 OAVO wird jeweils der ganzzahlige nicht gerundete Prozentsatz bzw. Fehlerindex zugrunde gelegt.

Für die Bewertung in den modernen Fremdsprachen ist der „Erlass zur Bewertung und Beurteilung von schriftlichen Arbeiten in allen Grund- und Leistungskursen der neu beginnenden und fortgeführten modernen Fremdsprachen in der gymnasialen Oberstufe, dem beruflichen Gymnasium, dem Abendgymnasium und dem Hessenkolleg“ vom 7. August 2020 (ABl. S. 519) zugrunde zu legen. Demnach erfolgt die Bewertung und Beurteilung mit der Maßgabe, dass lediglich bei der Ermittlung des Prüfungsergebnisses (Note) aus Prüfungsteil 1 und 2 gerundet wird.

Darüber hinaus sind die Vorgaben der Erlasse „Hinweise zur Vorbereitung auf die schriftlichen Abiturprüfungen (Abiturerlass)“, „Hinweise zur Vorbereitung auf die schriftlichen Abiturprüfungen im beruflichen Gymnasium (fachrichtungs-/ schwerpunktbezogene Fächer) (Abiturerlass BG)“ und „Durchführungsbestimmungen zum Landesabitur“ in der für den Abiturjahrgang geltenden Fassung zu beachten.

Als Kriterien für die Bewertung und Beurteilung dienen unter Beachtung der Zielsetzung der gymnasialen Oberstufe nach § 1 Abs. 2 OAVO neben dem Inhaltlichen auch die in den Kerncurricula genannten überfachlichen Kompetenzen, insbesondere die Sprachkompetenz und Wissenschaftspropädeutik; dies zeigt sich u.a. in qualitativen Merkmalen wie Strukturierung, Differenziertheit, (fach-)sprachlicher Gestaltung und Schlüssigkeit der Argumentation.

Im Fach Technische Informatik besteht die Prüfungsleistung aus der Bearbeitung eines Vorschlags, wofür insgesamt maximal 100 BE vergeben werden können. Ein Prüfungsergebnis von **5 Punkten (ausreichend)** setzt voraus, dass mindestens 45% der zu vergebenden BE erreicht werden. Ein Prüfungsergebnis von **11 Punkten (gut)** setzt voraus, dass mindestens 75% der zu vergebenden BE erreicht werden.

Gewichtung der Aufgaben und Zuordnung der Bewertungseinheiten zu den Anforderungsbereichen

Aufgabe	Bewertungseinheiten in den Anforderungsbereichen			Summe
	AFB I	AFB II	AFB III	
1	9	19	23	**51**
2	7	13	7	**27**
3	12	10		**22**
Summe	**28**	**42**	**30**	**100**

Die auf die Anforderungsbereiche verteilten Bewertungseinheiten innerhalb der Aufgaben sind als Richtwerte zu verstehen.